Inf Retrieval (2009) 12:509–525
DOI 10.1007/s10791-008-9069-5

The ineffectiveness of within-document term frequency in text classification

W. John Wilbur · Won Kim

Received: 27 November 2007 / Accepted: 2 September 2008 / Published online: 21 September 2008

Abstract For the purposes of classification it is common to represent a document as a bag of words. Such a representation consists of the individual terms making up the document together with the number of times each term appears in the document. All classification methods make use of the terms. It is common to also make use of the local term frequencies at the price of some added complication in the model. Examples are the naïve Bayes multinomial model (MM), the Dirichlet compound multinomial model (DCM) and the exponential-family approximation of the DCM (EDCM), as well as support vector machines (SVM). Although it is usually claimed that incorporating local word frequency in a document improves text classification performance, we here test whether such claims are true or not. In this paper we show experimentally that simplified forms of the MM, EDCM, and SVM models which ignore the frequency of each word in a document perform about at the same level as MM, DCM, EDCM and SVM models which incorporate local term frequency. We also present a new form of the naïve Bayes multivariate Bernoulli model (MBM) which is able to make use of local term frequency and show again that it offers no significant advantage over the plain MBM. We conclude that word burstiness is so strong that additional occurrences of a word essentially add no useful information to a classifier.

Keywords Within-document frequency · Bag-of-words · Word burstiness

1 Introduction

The intuition behind the naïve Bayes multinomial model (MM), as pointed out by Lewis (1998), is the impression that "if 1 occurrence of a word is a good clue that a document belongs to a class, then 5 occurrences should be even more predictive." An alternative

W. J. Wilbur (✉) · W. Kim
National Center for Biotechnology Information,
National Library of Medicine, National Institutes of Health, Bethesda, MD, USA
e-mail: wilbur@ncbi.nlm.nih.gov

W. Kim
e-mail: wonkim@ncbi.nlm.nih.gov

approach is the naïve Bayes multivariate Bernoulli model (MBM) in which local term frequency is ignored. A number of comparisons between the MM and the MBM models have appeared in the literature (Kalt 1996; McCallum and Nigam 1998; Eyheramendy et al. 2003; Schneider 2003) and all but Kalt found an advantage for the MM model. These results seem to support the importance of local term frequency in classification. On the other hand one of the serious problems with the MM model that has been recognized is the fact that repeated occurrences of the same word within a document are far from independent (Church 1995; Lewis 1998; Rennie et al. 2003; Teevan and Karger 2003; Pavlov et al. 2004; Schneider 2005), even though independence is an assumption of the MM model. This phenomenon has been termed word burstiness according to which if a word appears once in a document, it is more likely to appear again. The natural response to this is to attempt a modification of the MM model to take such dependencies into account. This led (Rennie et al. 2003) to introduce a log transformation of the local term frequency into the MM model. In a different approach (Madsen et al. 2005) show that while the MM approach does not well model word burstiness, the Dirichlet compound multinomial model (DCM) does a good job of modeling word burstiness and of classifying documents. In an important further development Elkan (2006) has introduced what he terms an exponential-family approximation of the DCM (EDCM) which is a very accurate approximation to the DCM but with important simplifications. In all three models, MM, DCM, and EDCM, the frequency of the term within a document is an integral part of the model. The important question is whether better modeling of within document frequencies leads to better classification.

Our claim is that the use of local term frequency within a document actually does not, as a rule, enhance document classification. To support this claim we will give a simplified version of MM and EDCM that performs at the same level as MM and EDCM. As further support for the claim, we also consider support vector machines (SVM). We use the $TF \times IDF$ normalized vector representation proposed by (Rennie et al. 2003) and which they show gives excellent results for document classification. We find that one may ignore the TF factor and that just the IDF normalized vector produces about the same results. Our results are based on an examination of 28 different classification problems many of which have been previously used in studies of document classification.

To add further evidence to our assertion that local within document frequency does not generally enhance classification, we also present a new approach to the use of local frequency in classification. This is an elaboration of the naïve Bayes multivariate Bernoulli model (MBM) which we term a stacked MBM. In this approach each higher frequency level of terms within a document is captured by a separate MBM. With this approach there is no need for an assumption of independence of the multiple occurrences of a term within a document, as is necessary with the MM approach. By this means we are able to show that the higher level MBMs in the stacked MBM approach do capture information about the class of a document. However, the information captured at the higher levels is not independent of the base level or standard MBM and is unable to enhance the performance coming from the standard MBM.

The paper is organized as follows. In Sect. 2.1 we present a brief summary of the MM, SMM, DCM, and EDCM models as well as how we construct vectors for the SVM approach. In Sect. 2.2 we present the MBM and the stacked MBM developed for this study. In Sect. 3 we describe the classification problems and how the data is prepared. Section 4 describes the evaluation measures we use. In Sect. 5.1 we give results for the MM, DCM, EDCM, and SVM approaches. In Sect. 5.2 we present our analysis of the stacked MBM approach. Sections 6 and 7 contain the discussion and conclusions.

2 Models

2.1 MM, SMM, DCM, EDCM, and SVM

The bag-of-words approach represents a document d by $\{tf_{vd}\}_{v\in V}$ where tf_{vd} is the number of times v appears in the document d. We will also be interested in the simplified approach in which document d is represented by $\{\delta_{vd}\}_{v\in V}$ where

$$\delta_{vd} = \begin{cases} 1, & \text{if } v \in d \\ 0, & \text{if } v \notin d \end{cases} \tag{1}$$

We assume that there are a fixed number of classes $C = \{c_1, c_2, \ldots, c_m\}$. Then for a probabilistic approach we may apply Bayes theorem to write

$$p(c_i|d) = \frac{p(d|c_i)p(c_i)}{p(d)} \tag{2}$$

It is sufficient to estimate $p(d|c_i)$, $p(c_i)$ and $p(d)$ to produce an estimate of $p(c_i|d)$. Generally $p(c_i)$ can be estimated based on training data with reasonable accuracy and for some applications it is not even necessary to make such an estimate. The probability $p(d)$ in (2) is not dependent on any class labels and it is unnecessary to estimate it for a classification task. On the other hand the estimate of $p(d|c_i)$ is crucial. The methods MM, DCM, and EDCM all provide an estimate for $p(d|c_i)$ in terms of the representation $\{tf_{vd}\}_{v\in V}$ for d. We here briefly sketch these estimates and provide for MM and EDCM alternate simplified estimates based on the $\{\delta_{vd}\}_{v\in V}$ representation for d.

2.1.1 Naïve Bayes multinomial model (MM)

The Naïve Bayes' assumption applied to $\{tf_{vd}\}_{v\in V}$ yields the multinomial model (Mitchell 1997; Lewis 1998; McCallum and Nigam 1998). This approach assumes that the occurrence of a token which is an instance of a $v \in V$ is governed by a probability that is independent of where the token occurs throughout the documents of a particular class in the database and estimated by the relative frequency of the token with respect to all the other tokens appearing in documents of that class in the database. To avoid probabilities of zero, one commonly adds a smoothing parameter ε. The result is

$$p(v|c_i) = \frac{\varepsilon + \sum_{d\in c_i} tf_{vd}}{\varepsilon|V| + \sum_{v'\in V, d\in c_i} tf_{v'd}}. \tag{3}$$

Historically Laplace priors have been assumed (Mitchell 1997; McCallum and Nigam 1998) which corresponds to setting $\varepsilon = 1$ for all words. However, recent research (Zhang and Oles 2001; Madsen et al. 2005) has found a value of $\varepsilon = 0.01$ to give improved results. Based on this formula for the probability of a single token, the formula for a multinomial probability may be applied to calculate $p(d|c_i)$ for a classification task.

$$p(d|c_i) = |d|! \prod_{v\in d} \frac{p(v|c_i)^{tf_{vd}}}{tf_{vd}!} \tag{4}$$

where $|d|$ is the length of the document, i.e., the total number of tokens in the document or the sum of the tf_{vd}.

In order to obtain the simplified MM (SMM) we replace tf_{vd} by δ_{vd} everywhere in Eqs. 3 and 4. The result is

$$p(v|c_i) = \frac{\varepsilon + n_{c_i v}}{\varepsilon|V| + \sum_{v' \in V} n_{c_i v'}} \tag{5}$$

where $n_{c_i v'}$ is the number of documents containing the term v' among all training documents from class c_i and

$$p(d|c_i) = n_d! \prod_{v \in d} p(v|c_i) \tag{6}$$

where n_d is the number of unique term types in document d. We note that essentially this simplified model was proposed and studied by (Schneider 2005) in a somewhat different context. Ideally one would determine an optimal value of ε for each classification problem. But to keep the amount of computation manageable, in all our applications of both (3) and (5) we use the value $\varepsilon = 0.01$.

2.1.2 Dirichlet compound multinomial model (DCM)

For the DCM distribution the probability of a document d belonging to a class c_i is determined by a vector of parameters $\boldsymbol{\alpha}^{c_i} = \{\alpha_v^i\}_{v \in V}$ (there will be such a parameter vector for each class c_i). The resulting probability for document d is

$$p(d|c_i, \boldsymbol{\alpha}^{c_i}) = \frac{\Gamma(s^i)}{\Gamma(s^i + |d|)} |d|! \prod_{v \in d} \frac{\Gamma(tf_{vd} + \alpha_v^i)}{tf_{vd}! \Gamma(\alpha_v^i)} \tag{7}$$

where $|d|$ is the length of the document and s^i is the sum of parameters α_w^i, i.e. $s^i = \sum_{v \in V} \alpha_v^i$. Given a class c_i, the parameter vector $\boldsymbol{\alpha}^{c_i}$ can be estimated from a training collection of documents D^{c_i} belonging to the class. The parameter vector $\boldsymbol{\alpha}^{c_i}$ is the maximum-likelihood solution which maximizes $\sum_{d \in D^{c_i}} \log(p(d|c_i, \boldsymbol{\alpha}^{c_i}))$. There exists no closed-form solution. An iterative gradient descent optimization method can be used to estimate the vector $\boldsymbol{\alpha}$ by computing the gradient of the DCM log likelihood. Two bound inequations are used with the gradient, leading to the update

$$\alpha_v^{new} = \alpha_v^{old} \frac{\sum_{d \in D^{c_i}} \Psi(tf_{vd} + \alpha_v^i) - \Psi(\alpha_v^i)}{\sum_{d \in D^{c_i}} \Psi(tf_{vd} + \sum_{v' \in V} \alpha_{v'}^i) - \Psi(\sum_{v' \in V} \alpha_{v'}^i)} \tag{8}$$

Here the function Ψ is the digamma function. In order to avoid zero α_v^i values the final solution is smoothed by adding 0.01 times the smallest nonzero α_v^i to all the $\alpha_{v'}^i$ values. What we have given here is essentially taken from (Madsen et al. 2005). For further details we refer the reader to (Minka 2003; Madsen et al. 2005).

2.1.3 Exponential-family approximation to the DCM (EDCM)

Elkan (2006) has derived a new family of distributions that is a close approximation to the DCM distributions and yet constitutes an exponential family, unlike DCM. He has used the EDCM distribution to obtain insights into the properties of the DCM distribution it approximates and has presented an algorithm for EDCM maximum-likelihood training that is many times faster than the corresponding method for the DCM distribution. Elkan

observes that for a typical case of document classification the DCM parameters satisfy $\alpha_v^i \ll 1$ for almost all $v \in V$. Since it is known that

$$\lim_{\alpha \to 0} \frac{\Gamma(x+\alpha)}{\Gamma(\alpha)} - \Gamma(x)\alpha = 0 \quad \text{for } x \geq 0 \tag{9}$$

then substituting into (7) based on the approximation suggested by (9) and using the fact that the local frequency tf_{vd} is an integer and $\Gamma(tf_{vd}) = (tf_{vd}-1)!$, the EDCM distribution is given by

$$p(d|c_i, \boldsymbol{\beta}^{c_i}) = \frac{\Gamma(s^i)}{\Gamma(s^i + |d|)} |d|! \prod_{v \in d} \frac{\beta_v^i}{tf_{vd}} \tag{10}$$

For clarity, the parameters $\boldsymbol{\beta}^{c_i}$ are used instead of $\boldsymbol{\alpha}^{c_i}$ in (10) for the EDCM parameters. As for the DCM a maximum likelihood estimate for the parameter vector $\boldsymbol{\beta}^{c_i}$ can be made based on a set of training documents D^{c_i} belonging to the class c_i. Such parameters maximize the concave function $\sum_{d \in D^{c_i}} \log\{p(d|c_i, \boldsymbol{\beta}^{c_i})\}$. The solution may be obtained from setting the partial derivatives of the log-likelihood to zero yielding

$$\beta_v^i = \frac{n_{c_i v}}{\sum_{d \in D^{c_i}} \Psi(s^i + |d|) - |D^{c_i}| \Psi(s^i)} \tag{11}$$

where $|D^{c_i}|$ is the total number of documents in the set D^{c_i} and $n_{c_i v}$ is the number of documents containing the term v in the set D^{c_i}. Summing the Eq. 11 over all v yields s^i on the left and the equation

$$s^i = \frac{\sum_{v \in V} n_{c_i v}}{\sum_{d \in D^{c_i}} \Psi(s^i + |d|) - |D^{c_i}| \Psi(s^i)}. \tag{12}$$

Since the only unknown in this equation is s^i, it can be solved numerically. Once s^i is known, the β_v^i can be directly computed from (11). As a practical matter we have found it quite useful to first solve for the EDCM model and use the resulting $\boldsymbol{\beta}^{c_i}$ as a starting point in computing the $\boldsymbol{\alpha}^{c_i}$ for the DCM model.

Again we smooth the parameters $\boldsymbol{\beta}^{c_i}$ by adding 0.01 times the smallest non-zero fitted β_v^i to all values as in the DCM model (Elkan 2006).

The EDCM is actually closely related to SMM, which can be seen by noting that

$$p(v|c_i) = \beta_v^i / s^i. \tag{13}$$

Thus we can rewrite the right side of (10) as

$$\left[\frac{|d|! \Gamma(s^i)}{n_d! \Gamma(s^i + |d|)} \prod_{v \in d} \frac{s^i}{tf_{vd}} \right] n_d! \prod_{v \in d} p(v|c_i). \tag{14}$$

Note that the EDCM probability is identical to the SMM probability given in (6) except for the bracketed coefficient which varies with the class only through s^i. We believe it is the unusual case where such an s^i provides information crucial to classification and find empirical support for this in the results that we report comparing EDCM and SMM.

2.1.4 Support vector machine (SVM)

In their classification study Rennie et al. (2003) reported their best results using a support vector machine applied to data prepared as vectors of $TF \times IDF$ term weights (Salton

1989; Baeza-Yates and Ribeiro-Neto 1999; Witten et al. 1999). The *TF* or local weight is produced as a log transformation of the local frequency

$$TF_{vd} = \log(1 + tf_{vd}) \tag{15}$$

which dampens the effect of local counts. The *IDF* or global weight is produced by the relatively standard

$$IDF_v = \log(|D|/n_v) \tag{16}$$

where $|D|$ is the total number of documents in the training set and n_v is the number of documents containing the word v in the set D. They multiply (15) and (16) in the standard approach and then normalize the result to obtain the final representation for a document. Thus the weight for an individual term v in a document d is

$$wt_v = \frac{TF_{vd} \cdot IDF_v}{\sqrt{\sum_{v' \in V} (TF_{v'd} \cdot IDF_{v'})^2}} \tag{17}$$

and the document is represented as $\{wt_v\}_{v \in V}$. We will refer to this form as the TF–IDF SVM. We will compare the TF–IDF SVM with the result of ignoring the local frequency of a term within a document, or equivalently setting

$$TF_{vd} = \begin{cases} 1, & \text{if } tf_{vd} > 0 \\ 0, & \text{if } tf_{vd} = 0 \end{cases} \tag{18}$$

in place of (15). The SVM that results from (18), (16), and (17) we will refer to as the IDF SVM. For most of our SVM experiments we have used SVM$^{\text{light}}$ with linear kernel and default parameters. Only for the TREC data have we used a new method based on a linear kernel and proposed by Joachims (2006) as of suitable time complexity for large data sets.

2.2 MBM and stacked MBM

Here Eq. (2) retains its importance and our goal is to define the quantity $p(d|c_i)$. For MBM this is based on the simple document representation $d = \{\delta_{vd}\}_{v \in V}$. However, for the stacked approach the bag of words representation $d = \{tf_{vd}\}_{v \in V}$ is used.

2.2.1 Multivariate Bernoulli model (MBM)

Here it is assumed that each term $v \in V$ occurs in a document, or does not occur, with a probability dependent only on that document's class and independent of the occurrence of any other term. Our approach is to define

$$n_{c_i v} = \sum_{d \in D^{c_i}} \delta_{vd} \tag{19}$$

and

$$n_v = \sum_{d \in D} \delta_{vd}. \tag{20}$$

Further set

$$f_i = |D^{c_i}|/|D| \tag{21}$$

Then we set

$$p_b(v|c_i) = \frac{\varepsilon f_i n_v + n_{c_i v}}{\varepsilon f_i |D| + |D^{c_i}|} \tag{22}$$

where $|D^{c_i}|$ denotes the number of documents in class c_i and ε is a small positive smoothing factor which throughout this work we take to be 0.01. We use the subscript b here to distinguish the probabilities discussed here from those of the MM. The multivariate Bernoulli document probability has a factor for each term

$$p_b(d|c_i) = \prod_{v \in V} p_b(v|c_i)^{\delta_{vd}} (1 - p_b(v|c_i))^{1-\delta_{vd}} \tag{23}$$

2.2.2 Stacked MBM (StMBM)

We introduce a new model which we term a *stacked* MBM model based on the equations

$$p(tf_v \geq k|c_i) = p(tf_v \geq 1|c_i)p(tf_v \geq 2|tf_v \geq 1 \wedge c_i)\ldots p(tf_v \geq k|tf_v \geq k-1 \wedge c_i) \tag{24}$$

$$\begin{aligned} p(tf_v = k|c_i) = {} & p(tf_v \geq 1|c_i)p(tf_v \geq 2|tf_v \geq 1 \wedge c_i)\ldots p(tf_v \geq k|tf_v \geq k-1 \wedge c_i) \\ & [1 - p(tf_v \geq k+1|tf_v \geq k \wedge c_i)] \end{aligned} \tag{25}$$

In order to apply this model we choose a local frequency limit, *f_limit*, beyond which we do not consider the counts to go. Then if a term occurs at a local frequency greater than this limit we set it to *f_limit* for purposes of our computations. Then for any term v and for any $k, 0 \leq k \leq f\_limit$ we define the counts

$$n_{c_i v}(k) = |\{d \in D^{c_i} | tf_{vd} \geq k\}| \tag{26}$$

and

$$n_v(k) = |\{d \in D | tf_{vd} \geq k\}|. \tag{27}$$

Then generalizing (22) and making use of f_i defined in (21), the elemental probabilities for $1 \leq k \leq f\_limit$ are given by

$$p(tf_v \geq k|tf_v \geq k-1 \wedge c_i) = \frac{\varepsilon f_i n_v(k) + n_{c_i v}(k)}{\varepsilon f_i n_v(k-1) + n_{c_i v}(k-1)}. \tag{28}$$

Now define the probabilities

$$q_{vd} = \begin{cases} p(tf_v \geq f\_limit|c_i), tf_{vd} \geq f\_limit \\ p(tf_v = tf_{vd}|c_i), tf_{vd} < f\_limit \end{cases}. \tag{29}$$

Then we may compute the probability of a document

$$p_s(d|c_i) = \prod_{v \in V} q_{vd} \tag{30}$$

where we use the subscript s to denote that this is a stacked MBM estimate.

3 Data sources and preparation

The databases we studied are:

MED[Heart]: MEDLINE® (McEntyre and Lipman 2001) documents that contained any MeSH® terms (Section 2004) that are below or equal to the term "Heart" in the MeSH

hierarchy. This set consists of 240,000 MEDLINE documents and MeSH terms are removed from the document representation. We will refer to this set as MED[Heart]. We considered the ten most frequent MeSH terms in the MED[Heart] set: human, animal, male, myocardium, female, heart, middle aged, adult, heart ventricles and myocardium/metabolism. We studied the binary classifications for each of these MeSH terms in the MED[Heart] set.

REBASE: The version of REBASE (a restriction enzyme database) we study here consists of 3,048 documents comprising titles and abstracts mostly taken from the research literature. These documents are all contained in MEDLINE. We have applied naïve Bayes (MBM) to learn the difference between REBASE and the remainder of MEDLINE and extracted the top scoring 100,000 documents from MEDLINE that lie outside of REBASE. We refer to this set as NREBASE. These are the 100,000 documents which are perhaps most likely to be confused with REBASE documents. We study the distinction between REBASE and NREBASE.

MDR dataset: The MDR dataset contains information from CDRHs (Center for Device and Radiological Health) device experience reports on devices which may have malfunctioned or caused a death or serious injury. The reports were received under both the mandatory Medical Device Reporting Program (MDR) from 1984 to 1996, and the voluntary reports up to June 1993. The database currently contains 620,119 reports that are divided into three disjoint classes: malfunction, death and serious injury. We studied the binary classifications for each of the three classes in the MDR set. The MDR set was used by (Eyheramendy et al. 2003) to study naïve Bayes models.

20 Newsgroups: A collection of messages, from 20 different newsgroups, with one thousand messages from each newsgroup. The data set has a vocabulary of 64,766 words. This data set has been studied by (McCallum and Nigam 1998; Eyheramendy et al. 2003; Rennie et al. 2003; Madsen et al. 2005; Schneider 2005).

Industry Sector: The Industry Sector data set contains 9,555 documents distributed in 104 classes. The data set has a vocabulary of 55,056 words. This data set has been studied by (McCallum and Nigam 1998; Rennie et al. 2003; Madsen et al. 2005).

WebKB: This data set (Craven et al. 1998) contains web pages gathered from university computer science departments. These pages are divided into seven categories: student, faculty, staff, course, project, department and other. We study the assignment of each of these category terms to documents as an independent binary decision. We do not exclude stop words to follow the suggestion of McCallum and Nigam (1998). This data set has been studied by McCallum and Nigam (1998) and Schneider (2005).

Reuters 21578: The ModApte train/test split of the Reuters 21578 Distribution 1.0 data set consists of 12,902 Reuters newswire articles in 135 overlapping topic categories. Following several other studies, we build binary classifiers for each of the ten most populous classes.

Because the above databases tend to have relatively short documents, we also studied the full-text document corpus that was assembled for the TREC 2007 Genomics Track:

2007 TREC Genomics data: The documents in this corpus came from the Highwire Press (www.highwire.org) electronic distribution of journals and there were slightly over 162,000 documents in the corpus from 49 genomics-related journals. Here we only considered journal articles that also appeared in the MEDLINE database and this set consists of 160,248 articles. We will refer to this set as TREC_GEN. We considered the 100 most frequent MeSH terms in the TREC_GEN set. The frequency of the MeSH terms varied from a high of 85,222 for the MeSH term "Animal" to a low of 3,565 for the MeSH term

Table 1 Document length is computed as the number of white space separated tokens in a document. Average, standard deviation, minimum and maximum document lengths over each database we studied are given here

	Average	Standard deviation	Min	Max
MED[Heart]	209.5	106.8	8	999
REBASE	206.6	79.9	33	795
MDR	42.307	38	0	403
20 Newsgroup	195.3	325.7	2	12741
Industry Sector	606.2	878.5	15	36655
WebKB	310.5	896.6	0	54580
Reuters	76.9	95.6	0	937
TREC_GEN	6066.85	2608.69	11	48697

"Cattle". We studied the binary classification problem corresponding to each of these 100 MeSH terms in the TREC_GEN set.

We processed the text of MED[Heart], REBASE, MDR, Reuters 21578, and TREC_GEN as follows:

All alphabetic characters are lowercased.
No stemming is done.
All non-alphanumeric characters are replaced by blanks.
All single nonstop terms and all adjacent pairs of nonstop terms without punctuation between are extracted from all documents to represent the initial V. There are two exceptions here: (1) for Reuters we only used single words and not pairs in order that our results might be more comparable to results reported by others (McCallum and Nigam 1998); (2) for TREC_GEN we processed the text with word pairs and singles, marked as TREC_GEN (D) and we also processed the text with just single words as features, marked as TREC_GEN (S).

For the Industry Sector, 20 Newsgroup and WebKB data, we used the Rainbow toolbox (McCallum 1996) to extract terms. The reason we used the Rainbow toolbox to extract the features is that we might be able to compare our results more fairly with the previously published results (McCallum and Nigam 1998; Rennie et al. 2003; Madsen et al. 2005), etc. Table 1 shows the document length characteristics of the databases we studied.

4 Evaluation

We use two different measures of performance. For the multi-class classification we use the accuracy which we compute as the fraction of all test cases that are correctly classified. This is the measure that we have most commonly seen applied to multi-class problems in the literature. We apply this measure for results on WebKB, Industry Sector and 20 Newsgroups.

For binary classification problems, all our methods attempt to rank the documents in what we might call the c_+ class above the documents we might call the c_- class. In this setting it is common and convenient to score the results as a precision-recall break-even point (BEP). Using BEP, the performance of the models is compared on the MeSH, REBASE, MDR, Reuters, and TREC_GEN collections. Both micro- and macro-averaged versions of BEP are given.

In all cases we train the algorithms on a training set and test on a held out test set. In the cases of data from MEDLINE and MDR the whole set of labeled data is randomly divided

into three parts and training takes place on two of these and testing on the third. In the case of Reuters 21578 there is a train/test division already defined for the data and we use it. For WebKB we hold out the data coming from Cornell for testing. For 20 Newsgroup, we split the data into 80/20 fractions for training and testing. For Industry Sector, it is split into halves for training and testing.

5 Results

5.1 MM, SMM, DCM, EDCM, and SVM

This section provides empirical evidence that incorporating word frequency in a document in most cases does not improve performance in the classification tasks.

The strongest trend one sees in examining the results of Fig. 1 is that the SMM results are almost identical to the DCM and EDCM results. The only exceptions are the WebKB data where DCM and EDCM slightly outperform SMM and the Industry Sector data where SMM is ahead of DCM and EDCM. The fact that DCM and EDCM produce almost identical results is expected based on the results of Elkan (2006) showing that these two approaches produce almost the same probabilities. Here we also see that SMM and MM are relatively close in performance most of the time with SMM appearing to have a small advantage. If we note that the data we report here are based on 27 individual classification problems (MeSH Terms 10, Reuters 10, MDR 3, REBASE 1, 20 Newsgroups 1, Industry

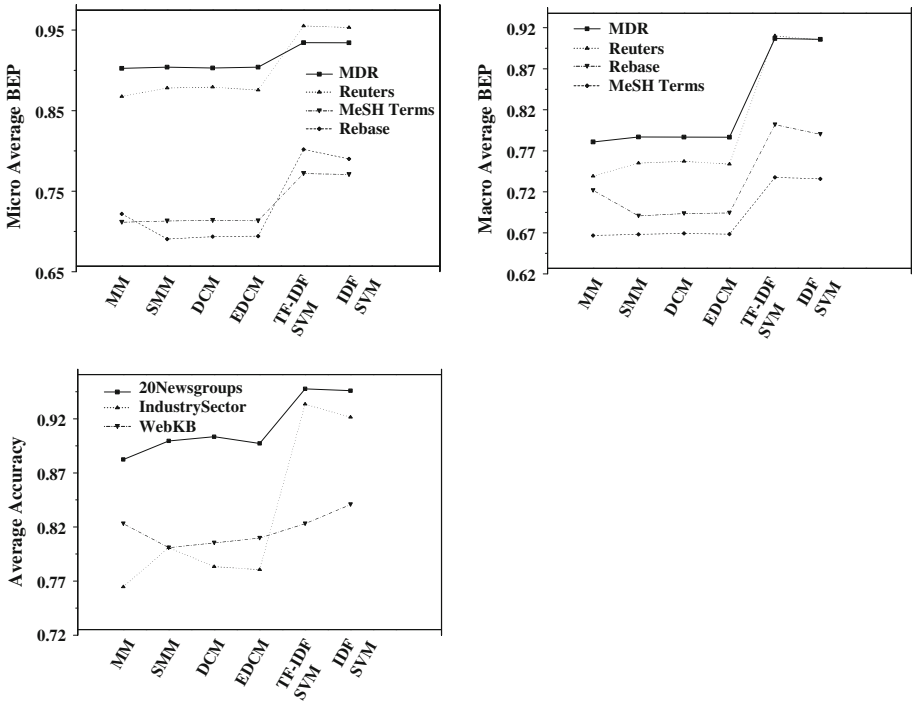

Fig. 1 A comparison of performance coming from the MM, SMM, DCM, EDCM, TF–IDF SVM, and IDF SVM models over the different binary and multiclass classification tasks we studied, except TREC_GEN

Sector 1, WebKB 1) and tally which classifier performs best on each problem we see (data not shown) that SMM wins 20 times, MM 6 times, and there is 1 tie. By the sign test this is statistically significant in favor of SMM. This observation that local frequency does not benefit in the MM model was also made by (Schneider 2005). Also in Fig. 1 one sees that TF–IDF SVM and IDF SVM appear to be very close in performance with TF–IDF SVM having a small advantage. Again if we consider the 27 individual classification tasks and tally the winners we see that TF–IDF SVM wins 12 times, IDF SVM wins 6 times, and there are 9 ties. This is not statistically significant by the sign test.

We have treated TREC_GEN as a separate problem and the results are given in Fig. 2. We treat TREC_GEN as a separate case because it consists of much longer documents than the other data sets and it may exhibit different characteristics. One of the challenges here is that the postings data for TREC_GEN involves over 25 million features and requires about 4 gigabytes of space. Each iteration over the data therefore requires substantial time. MM, SMM, and EDCM are simple and fast to compute and present no problem. DCM requires substantially more time, but is doable. For SVM we succeeded by using the improved method for linear kernals proposed in (Joachims 2006). In Fig. 2 results for MM and SMM show a slight advantage for MM over SMM, while for SVMs, TF–IDF SVM shows a small advantage over IDF SVM. Note that these advantages are slightly amplified for TREC_GEN (S) as opposed to TREC_GEN (D). However, TREC_GEN (D) IDF SVM is, on the scale of these differences, much better than TREC_GEN (S) TF–IDF SVM.

The foregoing results are produced using the full set of features contained in V. This is the approach used by (Madsen et al. 2005) for the DCM, by (Elkan 2006) for the EDCM, and by (Rennie et al. 2003) for the TF–IDF SVM. However, there is evidence to suggest that MM performance can be improved by feature selection (McCallum and Nigam 1998). In producing V we have already eliminated the high frequency functional words (stop terms). Beyond that there is reasonably good agreement in the classification community (Joachims 1997; Yang and Pedersen 1997; Craven et al. 1998; McCallum and Nigam 1998) that selecting features by their average mutual information with document class is useful and perhaps even the best technique. We will follow this approach for the multinomial model. In general the V will be taken as the terms with the m highest average mutual information values. Optimization of the MM method will be performed by varying m as in (McCallum and Nigam 1998). We examine 100 uniformly spaced values of m spanning the size of V to determine the optimal value of m in what follows. This feature

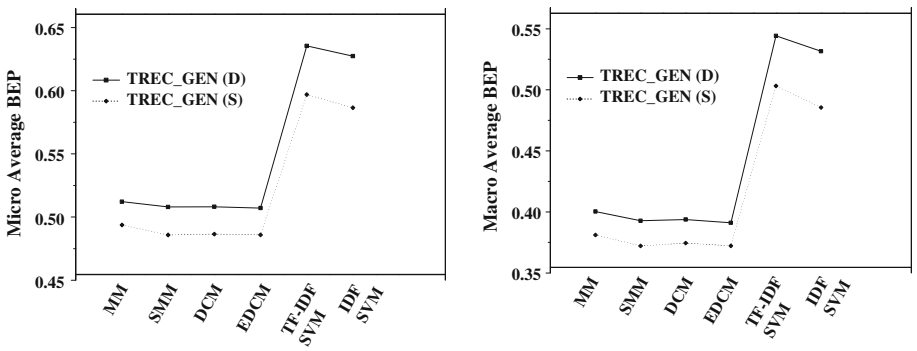

Fig. 2 A comparison of performance coming from the MM, SMM, DCM, EDCM, TF–IDF SVM, and IDF SVM models over the TREC_GEN data set

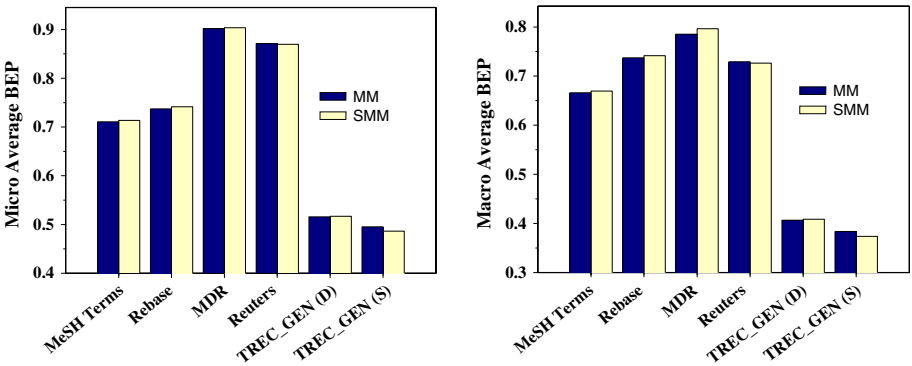

Fig. 3 A comparison of precision recall break even points coming from the MM and SMM models using only m terms selected by the mutual information determined by MM

selection is done on held out data to determine the optimal size m for MM performance on each classification problem. Based on this optimal m we then select the optimal V for MM based on the whole training set for that problem. We have then used that V to compute with both MM and SMM. We present results only for the binary classification problems as this seems sufficient to make our argument. Figure 3 presents the results.

The data in Fig. 3 shows that in general feature selection has improved performance for MM. Yet MM is still almost uniformly less effective than SMM. The one notable exception is the case of TREC_GEN (S) where MM clearly outperforms SMM. Here it is also notable that SMM on TREC_GEN (D) exceeds MM on TREC_GEN (S) by a relatively much larger factor than MM on TREC_GEN (S) exceeds SMM on TREC_GEN (S).

In order to summarize the comparison of MM and SMM we present Table 2. It is evident that SMM generally gives the superior performance. In the one case (REBASE) where the difference favors MM and approaches practical significance, we see that the advantage changes to SMM when optimal feature selection is used.

5.2 MBM and stacked MBM

The outstanding observation from the results of comparing MBM and StMBM, as given in Fig. 4, is that there is almost no performance difference in the two models. If we consider the 27 classification problems involving moderate length documents and let the average scores represent the Industry and 20 Newsgroup problems, then StMBM wins 15 times, MBM 5 times, and there are 7 ties. The sign test yields a p-value of 0.02 suggesting the there is a significant difference with the edge going to the StMBM model. However, the differences in scores are very small and unlikely to be of any practical significance. For the TREC_GEN data we see a pattern very similar to the comparison of MM and SMM in Fig. 3. StMBM and MBM are virtually identical in performance on (D), and both are substantially higher than either model on (S).

In an effort to gain a better understanding of what is happening here, we looked at the performance of just the higher level weights for local term frequencies of 2–5 coming from the StMBM model and compared the result to a random background model. The random scores are computed by taking the actual scores computed and shuffling them randomly among the documents in the test set and evaluating the performance 1,000 times and taking the average. From this same set of random scores we also estimate the 95% confidence

Table 2 Comparison of MM and SMM for the different datasets and conditions presented in Figs. 1–3

		Micro-average (%)		Macro-average (%)	
		MM	SMM	MM	SMM
BEP (without feature selection)	MED[Heart]	–	0.28	–	–
	REBASE	4.5	–	4.5	–
	MDR	–	0.11	–	0.65
	Reuters	–	0.12	–	0.22
	TREC_GEN (D)	0.79	–	1.8	–
	TREC_GEN (S)	1.64	–	2.4	–
Accuracy	Industry Sector	–	4.7		
	WebKB	2.7	–		
	20 Newsgroups	–	0.79		
BEP (optimal feature selection)	MED[Heart]	–	0.43	–	0.45
	REBASE	–	0.68	–	0.68
	MDR	–	0.22	–	0.15
	Reuters	0.11	–	–	0.41
	TREC_GEN (D)	–	0.19	–	0.49
	TREC_GEN (S)	1.85	–	2.67	–

In each case the better performer is marked with a percent improvement over the worse performer

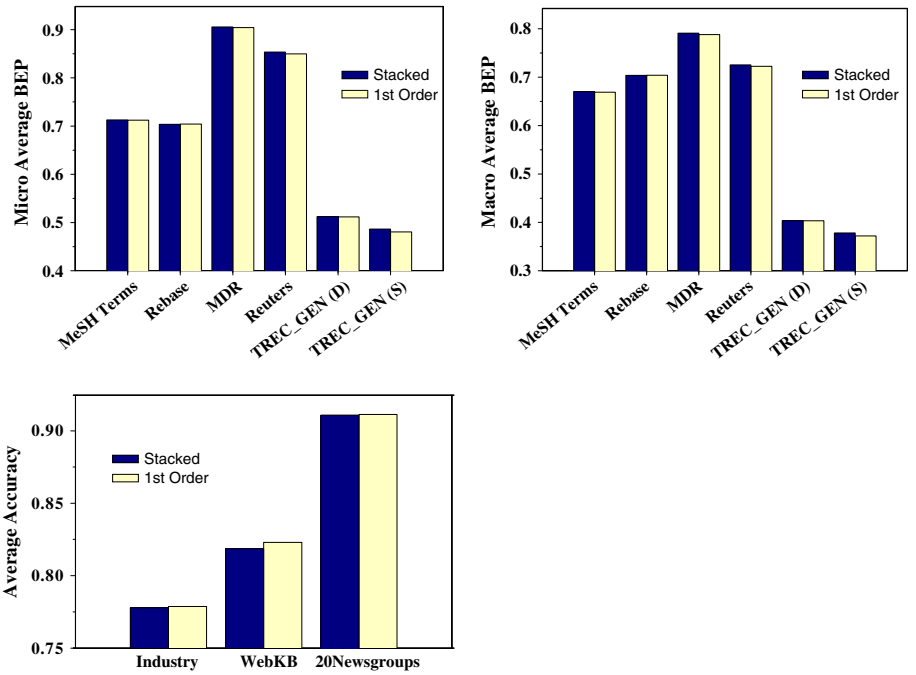

Fig. 4 For each problem data was held out of the training set and used to predict the optimal value of *f_limit* for the stacked model

limits. We found that for 21 cases out of 27 the higher order terms produced a classification result significantly above random. However, even when the performance was over 10 percentage points above random that did not translate into an improvement of a corresponding magnitude in the comparison of StMBM and MBM. For example the MeSH term "Animals" is classified with a BEP of 0.599 by the higher order terms while random performance is 0.480, but the StMBM achieves a BEP of 0.900 compared with a BEP of 0.899 for MBM on the same problem. Likewise the Reuters "earn" class is correctly labeled by the higher order terms with a BEP of 0.653 as opposed to a random result of 0.330. But we find StMBM performance is 0.939 while MBM achieves 0.937. Our interpretation of these results is that there is a strong dependency between the first order MBM weights and the higher order weights so that little is gained by adding the higher order weights.

6 Discussion

Perhaps the first issue to consider is whether our results with MM, EDCM, and SVM are consistent with the performance that others have reported. For MM we can compare our accuracy of 0.77 on the Industry Sector data with a figure of 0.78 by (Madsen et al. 2005) and 0.58 by (Rennie et al. 2003) (the latter uses no feature selection and does not optimize smoothing). Likewise for MM on the 20 Newsgroups data we obtain an accuracy of 0.88 which is to be compared with 0.85 for (McCallum and Nigam 1998), 0.85 for (Madsen et al. 2005), 0.86 for (Eyheramendy et al. 2003), and 0.85 for (Rennie et al. 2003). For DCM our accuracy of 0.783 on the Industry Sector data is to be compared with 0.806 for (Madsen et al. 2005) and our accuracy of 0.903 on the 20 Newsgroup data compares with 0.890 for (Madsen et al. 2005). Finally for the SVM we obtained an accuracy of 0.933 for the Industry Sector data and an accuracy of 0.948 for the 20 Newsgroup data. These figures are comparable to accuracies of 0.934 and 0.862, respectively, obtained by (Rennie et al. 2003) for a SVM. We present these comparisons as evidence that our implementations of MM, DCM, and SVM are competitive with what others have used. Thus we believe the explanation for our results does not lie in the implementations of the algorithms we use.

Our results suggest no benefit from using local frequency in the MM model on short to moderate length documents. Our SMM results are generally as good as our MM results. A natural question to ask is why then have most comparisons of MM shown better results than those for MBM. In answer to this question one may note that MM and even SMM are very different realizations of Naïve Bayes than MBM. It is quite believable that the differences between SMM and MBM, apart from local term frequencies, explain the differences that have been observed and reported by (Kalt 1996; McCallum and Nigam 1998; Eyheramendy et al. 2003; Schneider 2003). Further there are more or less effective ways to do feature selection and smoothing of probabilities in a Naïve Bayes model and what works best for MM may be different than what works best for MBM. The problem becomes reminiscent of trying to compare apples and oranges. How can one be sure one has equally optimized the two models to be compared? It is for this reason that we have used the SMM to compare with MM. SMM is much closer to MM in function and the method of optimization used is the same. We have applied this approach also to the comparison of TF–IDF SVM versus IDF SVM. In all of these comparisons the local term frequency is shown not to be responsible for the performance of the method because the simplified version without the local frequency factor matches the performance of its more complicated counterpart. This is based on consideration of performance on 27

classification problems, both binary and multiclass (Figs. 1–3). Applying the sign test to this data we find that SMM and IDF SVM are judged at least as good as MM, EDCM, DCM and TF–IDF SVM, respectively. Our important conclusion is that the advantages to using local term frequencies in these models are so small and so inconsistent over problems as to be without practical value.

Our investigation of MM, EDCM, DCM, and SVM has not yielded results which support the importance of local term frequency in document classification. However, this does not exclude the possibility that someone may invent a different model using local frequency that indeed shows that local frequency can make a substantial contribution to document classification. The fact that DCM actually provides a very accurate model for the behavior of local term frequencies (Madsen et al. 2005), and yet within that model such local frequencies do not seem to benefit classification, suggests that this may be difficult to do, but one cannot rule out the possibility on this basis alone. Because of this possibility we have developed what we have termed a stacked version of MBM which incorporates local term frequency and provides the possibility of multiple parameters to model the behavior of each separate term. A comparison of the results of MBM with StMBM shows that over the 27 problems with moderate length documents StMBM enjoys a statistically significant advantage by the sign test. However, a comparison of results on the individual problems shows that the differences are extremely small and again unlikely to have any practical significance. Because the advantage seen for StMBM is so small, we decided to examine the contribution of the higher order terms in StMBM to see if they in fact carry significant information. The results showed that in 21 of 27 cases classification based only on higher order terms of the StMBM are significantly above random. In many cases these higher order terms produce classification results far above random yet when combined with the first order MBM terms they produce almost no benefit over just the MBM. We believe this is best explained as due to a statistical dependency between the contributions of the higher order weights and those of the first order weights. In essence we are saying here that word burstiness is so strong that additional occurrences of a word add little information. If this is true it suggests that for moderate length documents no model will be able to use the local frequencies to advantage for classification.

The data we consider is composed mostly of short to moderate length documents (see Table 1). It is conceivable that longer documents might behave differently because there is much more opportunity for words to be emphasized by being used many times. In order to examine this possibility we included the TREC_GEN data set of full text scientific papers. We did this in two ways, as the set TREC_GEN (D) where two word phrases are included as features, and TREC_GEN (S) where only single words are used as features. While this was computationally challenging, it allowed for a more thorough comparison of MM and SMM on longer documents. On TREC_GEN (D) MM proved to have a slight advantage without feature selection. On the average over 100 classification problems, MM gave an improvement of 0.79% in the micro-averaged BEP compared with SMM. At the same time TF–IDF SVM gave an improvement of 1.3% in micro-averaged BEP compared with IDF SVM. However, when feature selection was applied SMM had a slight advantage over MM (see Table 2). On the same data we found that StMBM and MBM were essentially equivalent (Fig. 4). Thus we do not find support for the importance of local frequencies in classification on these long documents when two word phrases are included as features. The same computations and comparisons were made for TREC_GEN (S). With only single words used as features, we generally find that local frequency improves performance by a small amount. The largest of these improvements is for TF–IDF SVM over IDF SVM where the difference of the macro-averages is 3.6%. For the same comparison the

micro-averages differ by 1.8%. All the probabilistic models produce differences that are less than 3% for both micro- and macro-averages. Based on these computations we would say that one may gain a small advantage using local frequency in the (S) case, but this is arguably so small as to be of marginal importance. Further the improvement one may see in the (S) case by using local term frequency is much smaller than the improvement one sees in moving to the (D) representation without local frequency and this is true in all the models. Since we have only examined a single genre of data, however, it will be important to study more examples involving longer documents before reaching a final conclusion.

7 Conclusions

We have examined eight different data sets involving multiple classification problems with different characteristics and studied three different closely related model comparisons: SMM versus MM, DCM, and EDCM; IDF SVM versus TF–IDF SVM; and MBM versus StMBM. In all of this analysis we have failed to find evidence that there is substantial value for text classification in using the local frequencies of features (words or two word phrases) within documents. SMM is closely related to EDCM and its performance is almost identical with that of DCM and EDCM and usually as good as or better than MM. Differences between IDF SVM and TF–IDF SVM favor the use of local frequency, but are small and of marginal importance. Differences between MBM and StMBM favor StMBM and the use of local frequency but are so small as to be of no practical importance. These conclusions hold true in the one case of long documents (TREC_GEN) which we considered, but this issue requires further study for long documents.

Acknowledgements The authors are supported by the Intramural Research Program of the NIH, National Library of Medicine. The authors would like to thank the anonymous referees for numerous helpful comments which improved the work reported here.

References

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. Harlow, England: Addison-Wesley Longman Ltd.

Church, K. W. (1995). One term or two? In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Seattle, Washington: ACM Press.

Craven, M., DiPasquo, D., et al. (1998). Learning to extract symbolic knowledge from the World Wide Web. AAAI-98.

Elkan, C. (2006). Clustering documents with an exponential-family approximation of the Dirichlet compound multinomial distribution. In *23rd International Conference on Machine Learning*. Pittsburgh, Pennsylvania: ACM Press.

Eyheramendy, S., Lewis, D., et al. (2003). On the Naive Bayes model for text categorization. In *Ninth International Workshop on Artificial Intelligence & Statistics*, Key West, FL.

Joachims, T. (1997). A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. ICML-97.

Joachims, T. (2006). Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Philadelphia, PA, USA: ACM.

Kalt, T. (1996). A new probabilistic model of text classification and retrieval (1–9). Tech Report CIIR-TR-78, Univ. of Massachusetts.

Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. ECML.
Madsen, R. E., & Kauchak, D., et al. (2005). Modeling word burstiness using the Dirichlet distribution. In *22nd International Conference on Machine Learning*. Bonn, Germany: ACM Press.
McCallum, A. K. (1996). Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/∼mccallum/bow.
McCallum, A., & Nigam, K. (1998). A comparison of event models for naive bayes text classification. AAAI-98 Workshop on Learning for Text Categorization.
McEntyre, J., & Lipman, D. (2001). PubMed: Bridging the information gap. *Canadian Medical Association Journal, 164*(9), 1317–1319.
Minka, T. P. (2003). Estimating a Dirichlet distribution. Retrieved September 18, 2007, from https://research.microsoft.com/∼minka/papers/dirichlet/.
Mitchell, T. M. (1997). *Machine learning*. Boston: WCB/McGraw-Hill.
Pavlov, D., & Balasubramanyan, R., et al. (2004). Document preprocessing for naive Bayes classification and clustering with mixture of multinomials. KDD'04, Seattle, Washington.
Rennie, J. D. M., & Shih, L., et al. (2003). Tackling the poor assumptions of naive Bayes text classifiers. In *Twentieth International Conference on Machine Learning*, Washington, DC.
Salton, G. (1989). *Automatic text processing*. Reading, Massachusetts: Addison-Wesley Publishing Company.
Schneider, K.-M. (2003). A comparison of event models for Naive Bayes anti-spam e-mail filtering. EACL'03.
Schneider, K.-M. (2005). Techniques for improving the performance of Naive Bayes for text classification. In *Computational Linguistics and Intelligent Text Processing, 6th International Conference*, Mexico City, Springer.
Section, N. L. o. M. U. S. M. S. H. (2004). MeSH tree structures [electronic resource]/United States National Library of Medicine, National Institutes of Health, Medical Subject Headings. N. I. o. H. United States National Library of Medicine, U.S. National Library of Medicine, National Institutes of Health, Health & Human Services.
Teevan, J., & Karger, D. R. (2003). Empirical development of an exponential probabilistic model for text retrieval: Using textual analysis to build a better model. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, Toronto, Canada.
Witten, I. H., Moffat, A., et al. (1999). *Managing gigabytes*. San Francisco: Morgan-Kaufmann Publishers, Inc.
Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. *Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97)*.
Zhang, T., & Oles, F. J. (2001). Text categorization based on regularized linear classification methods. *Information Retrieval, 4*(1), 5–31. doi:10.1023/A:1011441423217.